Nonnegative Matrix Factorization with Gibbs Random Field Modeling

Shengcai Liao Zhen Lei Stan Z. Li*
Center for Biometrics and Security Research & National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences,
95 Zhongguancun Donglu, Beijing 100190, China.
{scliao, zlei, szli}@cbsr.ia.ac.cn

Abstract

In this paper, we present a Gibbs Random Field (GRF) modeling based Nonnegative Matrix Factorization (NMF) algorithm, called GRF-NMF. We propose to treat the component matrix of NMF as a Gibbs random field. Since each component presents a localized object part, as usually expected, we propose an energy function with the prior knowledge of smoothness and locality. This way of directly modeling on the structure of components makes the algorithm able to learn sparse, smooth, and localized object parts. Furthermore, we find that at each update iteration, the constrained term can be processed conveniently via local filtering on components. Finally we give a well established convergence proof for the derived algorithm. Experimental results on both synthesized and real image databases shows that the proposed GRF-NMF algorithm significantly outperforms other NMF related algorithms in sparsity, smoothness, and locality of the learned components.

1. Introduction

Nonnegative Matrix Factorization (NMF) [4] have been proved to be effective for learning object parts. It is a technique for decomposing a nonnegative data matrix into a product of two nonnegative matrices. NMF is widely used for discovering semantics in a given set of similar signals. The application areas include machine vision, video sequence analysis, spectral analysis, document clustering, and so on.

However, the classic NMF algorithm [4, 5] is not always successful in finding sparse and localized object parts [1]. In recent years several algorithms have been proposed to improve NMF for sparse and local components [6, 2, 7, 3, 10]. Most of them go for extracting sparse and local parts by utilizing some additional constrains, like sparse constrain [3], Lasso constrain [2, 7, 10], or orthogonal constrain [6, 10] on the factors. They might be successful in finding sparse and localized parts, however, they also get some noisy representations, or even break images into lots of nonsense tiny pieces when the quality of the input data is not so well. This is because none of them considers the prior structure of the learned components themselves. For example, two-dimensional images are always treated as concatenated one-dimensional signals, regardless of the mutual relationships of pixels among each other in original images.

There has been a newly proposed work [9] that considers the prior knowledge of the factors, but it assumes that the two NMF factors are independently determined by a Gaussian process. Under this assumption the relationship between the coefficient matrix and the component matrix is cut off, and it does not consider the underlying structure of the components as well.

In this paper, we try to directly model the prior object structure of the components into the NMF problem. We consider two prior knowledge about the components to learn: smoothness and locality. It is assumed that each component obeys smoothness constrain over neighboring elements, and it is locally concentrated. Taking this we apply the Gibbs Random Field (GRF) to effectively model them into the classic NMF problem, and accordingly we derive a new algorithm called GRF-NMF for learning sparse, local and smooth object parts.

The paper is structured as follows. In Section 2 we review the NMF technique, and introduce priors in a Bayesian framework. In Section 3 we model NMF with GRF, derive the GRF-NMF algorithm and show the corresponding convergence proof. Section 4 demonstrates experimental results on several data sets, and finally we summarize this paper in Section 5.

2. Bayesian Nonnegative Matrix Factorization with Priors

Given a nonnegative data matrix X of size $m \times n$, the classic NMF algorithm [4, 5] considers the problem of find-

[*] Stan Z. Li is the corresponding author.

ing two nonnegative matrices W and H, such that

$$X = WH + N, \tag{1}$$

where W is an $m \times r$ component matrix containing r components in columns, H is a coefficient matrix of size $r \times n$, and N is the residual noise matrix.

Consider that the noises are i.i.d. Gaussian distributed, with zero mean and variance of σ^2, then the likelihood of the factors W and H can be written as

$$P(X|W,H) = \frac{1}{(\sqrt{2\pi}\sigma)^{mn}} e^{-\frac{1}{2\sigma^2}\|WH-X\|_F^2}, \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenious-norm. Therefore, taking the negative log likelihood, the maximum likelihood (ML) estimate under nonnegative constrain can be formulated as the following classic NMF problem

$$\begin{aligned} &\min \ \ \|WH - X\|_F^2 \\ &s.t. \ \ W, H \geq 0, \end{aligned} \tag{3}$$

where the object function is known as the least squares likelihood or the Euclidian criterion [5]. Lee and Seung [5] shown that the NMF problem of Equ. (3) can be solved using two alternating multiplicative update rules as

$$W \leftarrow W \odot (XH') \oslash (WHH'), \tag{4}$$

$$H \leftarrow H \odot (W'X) \oslash (W'WH), \tag{5}$$

where P' is the transpose of matrix P, $\odot$ and $\oslash$ are elementwise multiplication (also known as the Hadamard product operator) and division operators between matrix respectively.

Recently, Schmidt and Laurberg proposed a maximum a posteriori (MAP) NMF method based on Gaussian process priors [9]. Using Bayes' rule, the posterior is formulated as

$$P(W,H|X) = \frac{P(X|W,H)P(W,H)}{P(X)}, \tag{6}$$

where $P(W,H)$ is a prior probability. Under the assumption that the factors of W and H are independent with each other, the prior could be further factorized as

$$P(W,H) = P(W)P(H). \tag{7}$$

As a result, the MAP estimate of Equ. (6) is equivalent to the following minimization problem

$$\min_{W,H \geq 0} -\ln P(X|W,H) + \alpha\phi_1(W) + \beta\phi_2(H), \tag{8}$$

where $\phi_1(W) \propto -\ln P(W)$, and $\phi_2(H) \propto -\ln P(H)$, with parameters $\alpha \geq 0$ and $\beta \geq 0$ balancing the constrains and the likelihood.

Many existing improved NMF methods [6, 2, 7, 3, 10, 9] can be viewed as special cases under this framework, with various priors related to W and H respectively to constrain the solution. However, the assumption of independent W and H might not properly be true, because at least they are correlated by a diagonal constant matrix Λ as a scale factor [4] such that WH is invariant under the transformation of

$$W \rightarrow W\Lambda, \ \ H \rightarrow \Lambda^{-1}H. \tag{9}$$

Therefore, to construct really effective constrains, we believe that W and H should be jointly modeled in the prior $P(W,H)$.

3. Gibbs Random Field Modeling of Nonnegative Matrix Factorization

It is known that in NMF each column of X and W represent a signal, then each entry of it may have its originally neighboring elements. Especially when input signals are two-dimensional images, each pixel has its spatially neighboring pixels. If the learned components represent object parts, as usually expected, they should obey some local smoothness constrain among neighboring pixels, as a normal image does. On the other hand, the size of each object part might not be large. The components are probably be some small localized parts of the object. In this work we mainly consider the two prior knowledge about the components, *i.e.* the smoothness and the locality. We apply the Gibbs Random Field (GRF) to effectively model them into the classic NMF problem. Accordingly we derive a new algorithm called GRF-NMF for learning sparse, local and smooth object parts.

3.1. Modeling Smoothness and Locality Priors

Let $\mathcal{S} = \{(i,k) \mid 1 \leq i \leq m, \ 1 \leq k \leq r\}$ be a set of sites on a regular lattice related to the component matrix W, with (i,k) indexing the i'th element of the k'th component. Consider a homogeneous neighborhood system on $\mathcal{S}$ such that $\mathcal{N} = \{\mathcal{N}_{ik} \cup \widetilde{\mathcal{N}}_{ik} \mid (i,k) \in \mathcal{S}\}$, where $\mathcal{N}_{ik} = \{(l,k) \mid l \in \mathcal{A}_i, (l,k) \in \mathcal{S}\}$, and $\widetilde{\mathcal{N}}_{ik} = \{(l,k) \mid l \in \mathcal{B}_i, (l,k) \in \mathcal{S}\}$. Both $\mathcal{N}_{ik}$ and $\widetilde{\mathcal{N}}_{ik}$ contain sites only in the same column of k, and their difference is that $\mathcal{N}_{ik}$ contains sites adjacent to (i,k), while $\widetilde{\mathcal{N}}_{ik}$ includes sites far away from (i,k). Note that $(l,k) \in \mathcal{N}_{ik}$ is equivalent to $l \in \mathcal{A}_i$. Define a $m \times m$ matrix A with $A_{il} = \mathbf{1}_{\mathcal{A}_i}(l)$, where $\mathbf{1}$ is the set indicator function, so that $l \in \mathcal{A}_i$ is equivalent to $A_{il} = 1$. Similarly, let B be a $m \times m$ matrix with $B_{il} = \mathbf{1}_{\mathcal{B}_i}(l)$, then the relationship among $\widetilde{\mathcal{N}}$, $\mathcal{B}$, and B is the same with that among $\mathcal{N}$, $\mathcal{A}$, and A. We call A and B *kernel matrices* of the neighborhood system. Also note that both A and B are symmetric matrices, because of the property of the neighborhood system.

For modeling the priors, we consider only pair-site (second order) cliques and define the following clique potentials

$$V_a(W_{ik}, W_{lk}) = \frac{1}{2}\alpha(W_{ik} - W_{lk})^2, \tag{10}$$

$$V_b(W_{ik}, W_{lk}) = \quad \beta W_{ik}W_{lk}, \tag{11}$$

where $\alpha \geq 0$ and $\beta \geq 0$ are constant weighting factors. The first potential function is a regularizer for $(l,k) \in \mathcal{N}_{ik}$, imposing the *a priori* smoothness constraint on the solution. On the other hand, the second one is a localization and sparsity term for $(l,k) \in \widetilde{\mathcal{N}}_{ik}$, indicates that the far away element W_{lk} is tend to be distinct from W_{ik}, making the component locally concentrated, and meanwhile it will naturally be sparse.

Let

$$\begin{aligned} f_k(W) =& \sum_i \left(\sum_{l\in\mathcal{A}_i} V_a(W_{ik}, W_{lk}) + \sum_{l\in\mathcal{B}_i} V_b(W_{ik}, W_{lk}) \right) \\ =& \sum_{i,l} \left(\frac{1}{2}\alpha A_{il}(W_{ik} - W_{lk})^2 + \beta B_{il} W_{ik} W_{lk} \right), \end{aligned} \tag{12}$$

and

$$g_k(H) = \sum_j H_{kj}^2, \tag{13}$$

then given H, we formulate the (conditional) prior energy function as

$$U(W|H) = \frac{1}{2}\sum_k f_k(W)g_k(H), \tag{14}$$

where $f_k(W)$ is considered as the prior energy of the k's component , and $g_k(H)$ can be viewed as a conditional weighting factor. In fact, $\sqrt{g_k(H)}$ is the L_2-norm of the k'th row of H, thus when $g_k(H)$ is larger, it means that in NMF formulation (*c.f.* Equ. (1)) the k's component is more important for reconstruction. Therefore we set the penalty on such component to be heavier accordingly.

Now the (conditional) prior distribution of W under the Gibbs distribution form can be written as

$$P(W|H) = \frac{1}{Z}e^{-U(W|H)}, \tag{15}$$

where Z is a normalizing constant. Assume that the prior distribution of the coefficient matrix is flat, then the joint prior distribution is

$$P(W,H) = P(W|H)P(H) \propto e^{-\frac{1}{2}\sum_k f_k(W)g_k(H)}. \tag{16}$$

The jointly modeled prior distribution considers both W and H, and the scale correlation between them. In this way, further combine with the likelihood distribution in Equ. (2), the posterior distribution will be

$$\begin{aligned} P(W,H|X) &\propto P(X|W,H)P(W,H) \\ &\propto e^{-\frac{1}{2\sigma^2}\|WH-X\|_F^2 - \frac{1}{2}\sum_k f_k(W)g_k(H)}. \end{aligned} \tag{17}$$

As a result, the MAP estimate of Equ. (17) can be found by minimizing the following GRF-NMF object function[1]

$$J(W,H) = \frac{1}{2}\|WH - X\|_F^2 + \frac{1}{2}\sum_k f_k(W)g_k(H). \tag{18}$$

And accordingly, the GRF-NMF problem is defined as

$$\begin{aligned} &\min \;\; J(W,H) \\ &s.t. \;\; W,H \geq 0, \;\; \sum_i W_{ik} = 1, \forall k, \end{aligned} \tag{19}$$

where the latter constrain is set for a unique solution, because $J(W,H)$ is invariant under the transformation of Equ. (9), which can be easily verified.

The object function of Equ. (18) is not convex with respect to both W and H. It is popularly solved by alternating updates. When H is fixed, as discussed above, minimizing $J(W,H)$ with respect to W will maximize the likelihood in the constrain of smooth, local, and sparse components, with α, β, and $g(H)$ balancing between them. On the other hand, when W is fixed, $f_k(W)$ can be viewed as a conditional weighting factor measuring the *degree of dissatisfaction* of the k'th component. Therefore the k'th row of H will be penalized more if $f_k(W)$ is larger. Consequently, minimizing $J(W,H)$ with respect to H will maximize the likelihood in the constrain of weighted L_2-norm of H, and it will make the solution of H conditionally sparse.

3.2. Algorithm

To solve the GRF-NMF problem of Equ. (19), we derive two alternating matrix-wise multiplicative update rules as

$$W \leftarrow W \odot (XH' + SG) \oslash (WHH' + TG), \tag{20}$$

$$H \leftarrow H \odot (W'X) \oslash (W'WH + FH), \tag{21}$$

where $S(W)$ and $T(W)$ are $m \times r$ nonnegative matrices computed as

$$S_{ik}(W) = 2\alpha \sum_{l\in\mathcal{A}_i} W_{lk}, \tag{22}$$

$$T_{ik}(W) = \alpha \sum_{l\in\mathcal{A}_i}(W_{ik} + W_{lk}) + \beta \sum_{l\in\mathcal{B}_i} W_{lk}, \tag{23}$$

and $F(W)$ and $G(H)$ are $r \times r$ diagonal nonnegative matrices defined as

$$F_{ik}(W) = \delta_{ik} f_k(W), \;\; G_{kj}(H) = \delta_{kj} g_k(H), \tag{24}$$

with δ_{ij} be the Kronecker delta. Matrices $F(W)$ and $G(H)$ are two diagonal weighting matrix, as analyzed in Section 3.1. The multiplicative update rules of Equ. (20)(21) are

[1] For convenience, σ in Equ. (17) is set to be 1, and α and β in Equ. (12) is rescaled accordingly.

similar with that of the classic NMF (*c.f.* Equ. (4)(5)), while more constrains are added to yield desired solution. For example, the term FH in the denominator of Equ. (21) constrains H to be conditionally sparse weighted by $F(W)$. Note that the update procedure only involves matrix operations, so we can efficiently process it with some matrix computing engine (*e.g.* MATLAB).

An interesting finding is that, according to Equ. (22), $S(W)$ can be rewritten as $S_{ik}(W) = 2\alpha \sum_l A_{il} W_{lk} = 2\alpha(AW)_{ik}$. Hence the matrix S is actually a result of local smooth filtering on each component respectively, with rows of matrix $\Gamma^S = 2\alpha A$ as the corresponding filter kernels. In this way, the update rule of Equ. (20) makes W locally smoothed. If the input signal is image, then each column of W represents component image. As a result, not only columns of W, but also rows of Γ^S can be reshaped as 2-D images. We denote the local valid part of the reshaped 2-D filter kernel from rows of Γ^S as $K^S_{\omega\times\omega}$ [2], with filter size ω indicating the affected local smoothing area. For example, it can be of the form

$$K^S_{3\times3} = \begin{bmatrix} 0 & \alpha & 0 \\ \alpha & 0 & \alpha \\ 0 & \alpha & 0 \end{bmatrix}, \quad K^S_{3\times3} = \begin{bmatrix} \alpha & \alpha & \alpha \\ \alpha & 0 & \alpha \\ \alpha & \alpha & \alpha \end{bmatrix} \tag{25}$$

for 4-neighborhood system and 8-neighborhood system respectively.

For computation of $T(W)$, we define two constant $m \times m$ matrices as

$$C_{il} = 1 - B_{il}, \;\; D_{il} = \delta_{il} \sum_p A_{ip}. \tag{26}$$

Then $T(W)$ can be reformulated as

$$\begin{aligned} T_{ik}(W) &= \frac{1}{2} S_{ik}(W) + \sum_l \left(\alpha D_{il} + \beta B_{il}\right) W_{lk} \\ &= \frac{1}{2} S_{ik}(W) - R_{ik}(W) + \alpha(DW)_{ik} + \beta \sum_l W_{lk}, \end{aligned} \tag{27}$$

where

$$R(W) = \beta C W. \tag{28}$$

Note that $R(W)$ is also a result of local smooth filtering on components, with rows of matrix $\Gamma^R = \beta C$ as the corresponding filter kernels. Therefore Equ. (27) shows that the computation of $T(W)$ only involves two local smooth filtering processing, with αDW be a diagonally rescaled version of W, and $\sum_l W_{lk}$ only needs to be computed once for each column. When dealing with images, $K^R_{\tau\times\tau}$ is defined as the local valid part of the reshaped 2-D filter kernel from rows of Γ^S, with filter size τ indicating the affected local smoothing area. It can be verify that $K^R_{\tau\times\tau}$ is actually a constant matrix with all elements to be β. It can also be inferred that τ is related with the size of the object parts represented by components.

Taking above computations, the GRF-NMF algorithm is detailed as follows.

```
Algorithm: GRF-NMF
 1. Input:
      (a) Nonnegative data matrix $X$.
      (b) The number of components $r$.
      (c) Max iterations $L$.
      (d) Constrain parameters $\alpha$ and $\beta$.
      (e) Neighborhood kernel matrices $A$ and $B$.
 2. Initialize:
      (1) Assign $W$ to a random positive matrix.
      (2) Set $H = W'X$.
 3. Iterate:
      For $t = 1$ to $L$
        (1) Compute $f$ and $g$ with Equ. (12)(13),
            and assign $F$ and $G$ with Equ. (24).
        (2) Compute $S$ and $R$ by filtering each component
            of $W$ with $\Gamma^S$ and $\Gamma^R$ respectively.
        (3) Compute $T$ with Equ. (27).
        (4) Update $H$ and $W$ by Equ. (20)(21), and
            normalize via Equ. (9) with $\Lambda_{kk} = 1/\sum_i W_{ik}$.
      End
 4. Output: $W, H$.
```

3.3. Proofs of Convergence

Theorem 1 *The object function of Equ. (18) is non-increasing under the update rule of Equ. (21).*

Theorem 2 *The object function of Equ. (18) is non-increasing under the update rule of Equ. (20).*

The above two theorems ensure that the GRF-NMF algorithm will converge after a certain number of alternating update iterations. The proofs are similar with [5], making use of auxiliary functions and a gradient decent scheme with a multiplicative update style for preserving non-negativity. The difference in this paper is that we are dealing with the case of updating the whole matrix simultaneously. Here we detail it with several following lemmas.

Before going on, we introduce the notation of *equally partitioned matrix* used in this paper. A matrix P of size $mn \times mn$ is called to be an equally partitioned matrix if it has the form

$$P = [P_{ij}] = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1n} \\ P_{21} & P_{22} & \cdots & P_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ P_{n1} & P_{n2} & \cdots & P_{nn} \end{bmatrix},$$

where all P_{ij} are matrices of the same size $m \times m$. And we further use the notation of P^{kl}_{ij} to index the row k and column l element of P_{ij}.

[2] Since the neighborhood system is homogeneous, $\Gamma^S = 2\alpha A$ can be determined by shifting $K^S_{\omega\times\omega}$

Lemma 1 *The gradient of the object function of Equ. (18) with respect to $\boldsymbol{h} = vec(H)$ is $\nabla_{\boldsymbol{h}} J(W,H) = vec(W'WH - W'X + F(W)H)$, and the corresponding Hessian matrix is $\nabla^2_{\boldsymbol{h}} J(W,H) = I_n \otimes (W'W + F(W))$, where $\otimes$ denotes the Kronecker product operator.*

Proof: It is easy to verify that

$$\frac{\partial J(W,H)}{\partial H_{kj}} = (W'WH)_{kj} - (W'X)_{kj} + f_k H_{kj}, \quad (29)$$

$$\frac{\partial^2 J(W,H)}{\partial H_{kj} \partial H_{\mu\nu}} = \delta_{j\nu}(W'W)_{k\mu} + \delta_{k\mu}\delta_{j\nu} f_k. \, \square \quad (30)$$

Lemma 2 *If $Q(H^t)$ is a diagonal equally partitioned matrix of the form*

$$Q^{k\mu}_{j\nu}(H^t) = \delta_{k\mu}\delta_{j\nu}(W'WH^t + F(W)H^t)_{kj}/H^t_{kj}, \quad (31)$$

then

$$\begin{aligned}\Phi(H,H^t) &= J(W,H^t) + (\mathbf{h} - \mathbf{h}^t)'\nabla_{\mathbf{h}} J(W,H^t) \\ &\quad + \frac{1}{2}(\mathbf{h} - \mathbf{h}^t)' Q(H^t)(\mathbf{h} - \mathbf{h}^t)\end{aligned} \quad (32)$$

is an auxiliary function for $J(W,H)$.

Proof: The proof is similar with that of [5], except that we prove the case of updating the whole matrix H in a while with Jacobi style. Here we only show one critical step that the matrix

$$M^{kj}_{\mu\nu}(H^t) = H^t_{kj}\left[Q(H^t) - \nabla^2_{\mathbf{h}} J(W,H^t)\right]^{kj}_{\mu\nu} H^t_{\mu\nu} \quad (33)$$

is positive semidefinite. In fact, according to Lemma 1 and Equ. (31)(33), we have

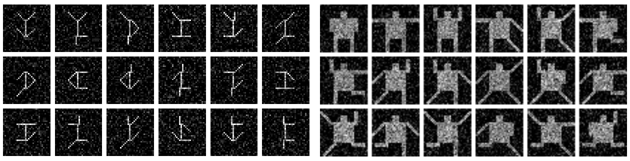
$$\begin{aligned}\mathbf{z}'M(H^t)\mathbf{z} &= \sum_{kj,\mu\nu} \mathbf{z}_{kj} M^{kj}_{\mu\nu}(H^t)\mathbf{z}_{\mu\nu} \\ &= \sum_{kj,\mu\nu} \mathbf{z}_{kj}\delta_{j\nu}\left[\delta_{k\mu}(W'WH^t)_{k\mu} - H^t_{kj}(W'W)_{k\mu}\right] H^t_{\mu\nu}\mathbf{z}_{\mu\nu} \\ &= \sum_{kj\mu} \left[\mathbf{z}^2_{kj}(W'W)_{k\mu}H^t_{\mu j}H^t_{kj} - \mathbf{z}_{kj}H^t_{kj}(W'W)_{k\mu}H^t_{\mu j}\mathbf{z}_{\mu j}\right] \\ &= \sum_{kj\mu} H^t_{kj}(W'W)_{k\mu}H^t_{\mu j}\left(\frac{1}{2}\mathbf{z}^2_{kj} + \frac{1}{2}\mathbf{z}^2_{\mu j} - \mathbf{z}_{kj}\mathbf{z}_{\mu j}\right) \\ &= \frac{1}{2}\sum_{kj\mu} H^t_{kj}(W'W)_{k\mu}H^t_{\mu j}(\mathbf{z}_{kj} - \mathbf{z}_{\mu j})^2 \geq 0. \; \square\end{aligned}$$

Proof of Theorem 1 Minimizing $\Phi(H,H^t)$ in Equ. (32) gives the update rule of

$$\mathbf{h}^{t+1} = \mathbf{h}^t - Q(H^t)^{-1}\nabla_{\mathbf{h}} J(W,H^t). \quad (34)$$

According to Lemma 2, $J(W,H)$ is nonincreasing under this update rule, since $\Phi(H,H^t)$ is an auxiliary function. Note that $H = unvec(\mathbf{h})$, thus Equ. (34) is equivalent to the matrix update rule of Equ. (21). $\square$

Theorem 2 can be proved in the same way, with some more details. We omit it here.



(a) Swimmer data set (b) Robot data set

Figure 1. Examples of disturbed images.

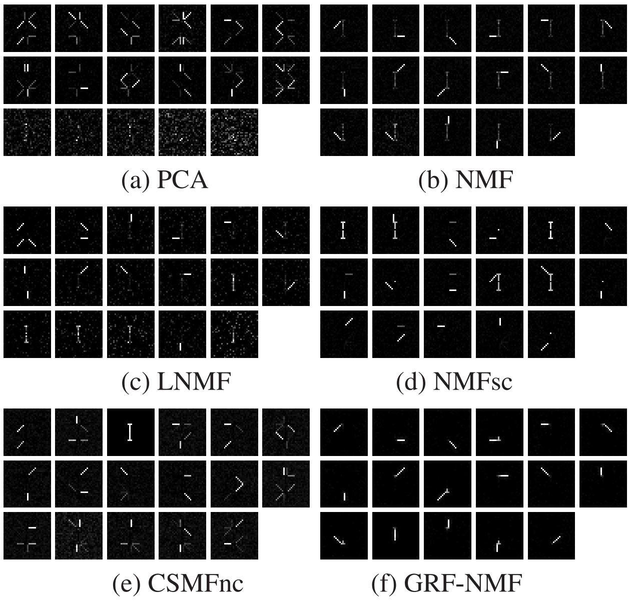
4. Experiments

To evaluate the performance of NMF related algorithms, we run experiments on several data sets containing noisy images, trying to recover the underlying local object parts. Several variants of NMF algorithms are compared, including LNMF [6], NMFsc [3], and CSMFnc [10]. In the experiments all NMF related algorithms are initialized with the same value whenever possible. To apply GRF-NMF algorithm with images, we use two 2-D smooth filters K^S and K^R to form the neighborhood system by shifting. For filter K^S we use an 8-neighborhood setting in the form of Equ. (25), and filter K^R is set to be a flat matrix of size $\tau \times \tau$ with all elements to be β.

4.1. Swimmer data set

The Swimmer data set contains 256 images of size 32×32 [1]. Each image is composed of five parts: an invariant part called "torso" in the middle and 4 "limbs" in 4 positions, each of which has 4 changing directions. Therefore all images can be generated additively by 17 distinct components. For each image, we add a Gaussian noise of zero mean and 0.2 standard variance. Fig. 1(a) shows some examples of the noisy images. The number of components to learn is set to be $r = 17$, and the maximum iterations for learning is set to be 300.

Fig. 2 demonstrates the learned components of all compared algorithms, and Fig. 3 displays the corresponding reconstructed images. From Fig. 2 we can see that only GRF-NMF algorithm correctly learns all 17 components from noisy data. It is the best representation for sparse, local, and smooth object parts. Meanwhile, Fig. 3(f) shows that most images are perfectly reconstructed by GRF-NMF. It is known that the classic NMF solution often contains ghost parts, as shown in Fig. 2(b). The LNMF algorithm improves NMF aiming to find local representation, but it is not successful here for factorizing noisy images (*c.f.* Fig. 2(c)), and the reconstruction is poor (*c.f.* Fig. 3(c)). Components learned by NMFsc is indeed sparse, but they have duplicated torso, and not all of the 17 components are separated alone. In literature only the CSMFnc [10] algorithm has successfully found all individual components in the Swimmer data. However, we find that CSMFnc is sensitive to initial values, and noises. As shown in Fig. 2(e), the CSMFnc solution contains ghost parts and duplicate components.

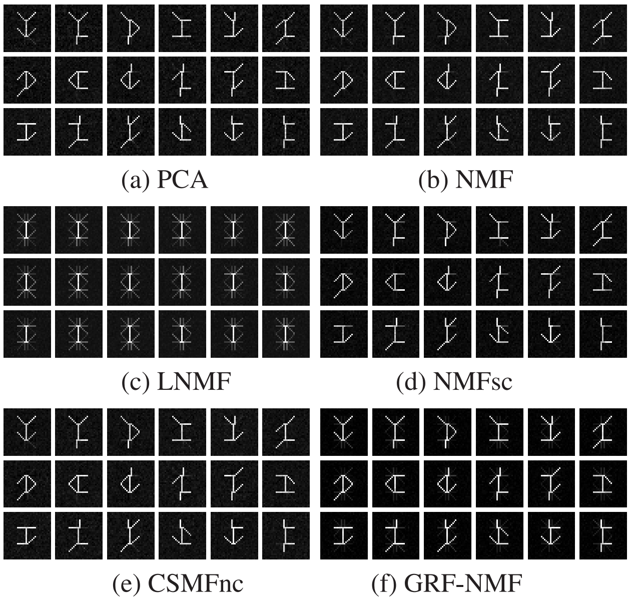


(a) PCA (b) NMF

(c) LNMF (d) NMFsc

(e) CSMFnc (f) GRF-NMF

Figure 2. Learned components on Swimmer data set. Parameters for NMFsc are $S_w = 0.8$ and $S_h = 0.5$. Parameters for CSMFnc are set to be $\alpha = 0.01, \beta = 0.01, \lambda = 0.01$, $N^w_{0,1} = 600$, $N^w_{0,2} = 200$, $N^h_{0,1} = 150$ and $N^h_{0,2} = 50$, as recommenced in [10]. Parameters for GRF-NMF are set to be $\alpha = 0.001, \beta = 0.01, \tau = 5$.



(a) PCA (b) NMF

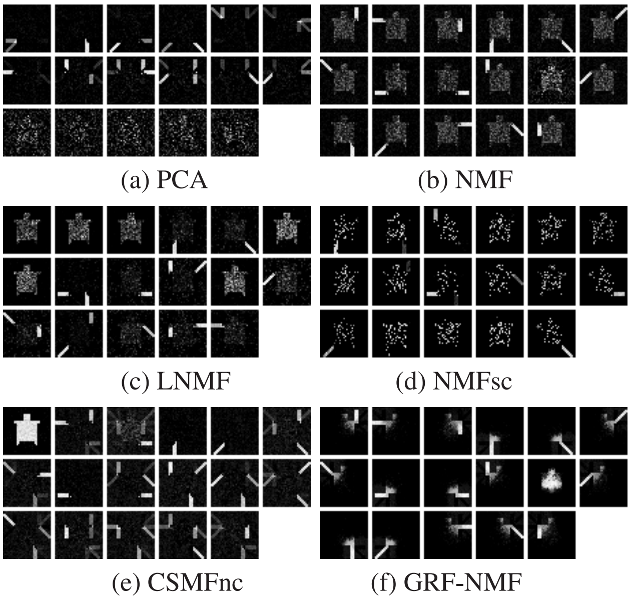
(c) LNMF (d) NMFsc

(e) CSMFnc (f) GRF-NMF

Figure 3. Some reconstructed images of Swimmer data using components of Fig. 2.

4.2. Robot Data Set

We also construct another synthesized data named "Robot" to evaluate the ability of NMF related algorithms for finding underling object parts, as shown in Fig. 1(b). The size of the Robot images is 35×35. Each image contains a common big body of size 15×15 in the center, and also a head of size 5×5. The arms and legs are 10 pixels in length and 3 pixels in width, with each leg has 3 directions to vary and each arm 5. Consequently there are total 225 images containing 17 different components, in which the body and the head part are counted as one invariant component. All images are generated by these components additively, and further disturbed with a Gaussian noise of zero mean and 0.3 standard variance. The number of components to learn is set to be $r = 17$, and the maximum iterations for learning is set to be 500.



(a) PCA (b) NMF

(c) LNMF (d) NMFsc

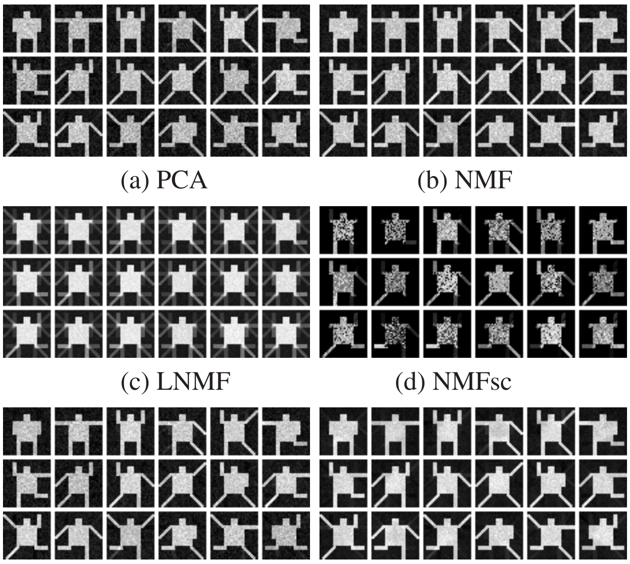
(e) CSMFnc (f) GRF-NMF

Figure 4. Learned components on Robot data set. Parameters for NMFsc and CSMFnc are the same as in Fig. 2. Parameters for GRF-NMF are set to be $\alpha = 0.001, \beta = 0.001$, and $\tau = 25$.

The results are demonstrated in Fig. 4 and Fig. 5, showing all learned components and the corresponding reconstructed images respectively. Again, the proposed GRF-NMF algorithm outperforms all other compared algorithms in component discovery and object reconstruction. As shown in Fig. 4, only GRF-NMF algorithm correctly finds out all 17 components from noisy images, though not precise enough. Besides, all components learned by GRF-NMF are smooth, sparse, and localized. Also notice from Fig. 4 that all NMF, LNMF, and NMFsc components contain heavy ghost effects. And the reconstruction of LNMF is dissatisfactory (*c.f.* Fig. 5(c)). The solution of CSMFnc algorithm (*c.f.* Fig. 4(e)) contains a full body-head part, but arms and legs are disorganized.

4.3. Face Images

Finally we use the FERET database [8] to evaluate the performance of NMF algorithms on real images. We select a subset of the FERET database that contains 540 face im-



(a) PCA (b) NMF

(c) LNMF (d) NMFsc

(e) CSMFnc (f) GRF-NMF

Figure 5. Some reconstructed images of Robot data using components of Fig. 4.

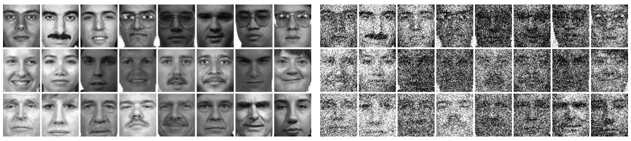
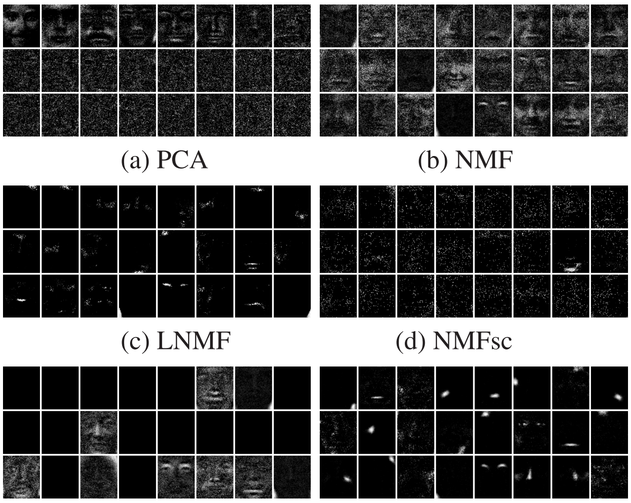


Figure 6. Examples of cropped faces (left) and their corresponding corrupted images (right) from FERET database.

ages from 270 subjects. Face images are cropped into 75 pixels high and 65 pixels wide, according to their eye coordinates. No preprocessing is used but a Gaussian noise of zero mean and 0.3 standard variance is added on each cropped image. Some examples of the cropped face images and their corresponding disturbed images are illustrated in Fig. 6. In the experiment, the number of components to learn is set to be 100. GRF-NMF is updated by only 500 iterations, while the maximum iterations of other NMF related algorithms are set to be 2000 for sufficient learning.

Fig. 7 and Fig. 8 demonstrate the learned components and their corresponding reconstructed images. It can be seen that GRF-NMF learns the best solution for representation of sparse, localized, and smoothed components. It finds meaningful face parts such as nose, mouth, eyes, eyebrows, and cheeks. All other results are noisy. The result of NMFsc contains nonsense sparse representation. The CSMFnc solution is over constrained. Result of LNMF is sparse, but the reconstruction is not so well. And LNMF takes long time to converge to meaningful result (more than 1000 iterations in our experiment).

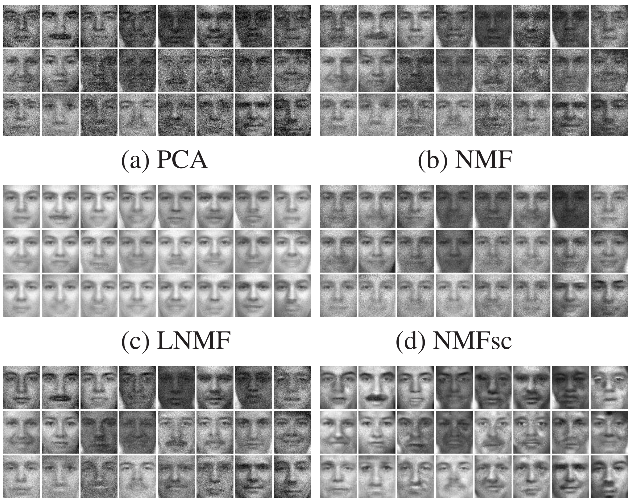


(a) PCA (b) NMF

(c) LNMF (d) NMFsc

(e) CSMFnc (f) GRF-NMF

Figure 7. Learned components on FERET database. Parameters for NMFsc are $S_w = 0.8$ and $S_h = 0.5$. Parameters for CSMFnc are set to be $\alpha = 0.1, \beta = 1, \lambda = 0.2$, $N_{0,1}^w = 3000$, $N_{0,2}^w = 1000$, $N_{0,1}^h = 300$ and $N_{0,2}^h = 100$. Parameters for GRF-NMF are set to be $\alpha = 0.01, \beta = 0.001$, and $\tau = 11$.



(a) PCA (b) NMF

(c) LNMF (d) NMFsc

(e) CSMFnc (f) GRF-NMF

Figure 8. Some reconstructed images of FERET database using components of Fig. 7.

4.4. Parameter Selection

The success of GRF-NMF does not depend on some carefully selected parameters. In fact, a range of parameter values can yield satisfactory results. In order to understand how parameters (*i.e.*, α, β, and τ) of GRF-NMF affect the factorization results, we give some additional experimental results with some other parameters for comparison. The results are illustrated in Fig. 9 and Fig. 10 for learned components and reconstructed images respectively.

First, from Fig. 9(a) ($\tau = 21$) and Fig. 9(b) ($\tau = 41$)

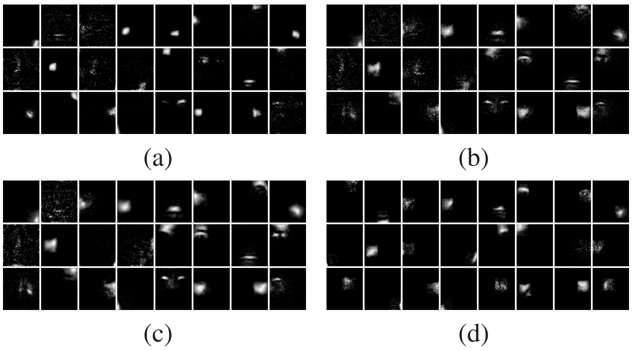


Figure 9. Learned components on FERET database by GRF-NMF with various parameters. (a) $\alpha = 0.01, \beta = 0.001, \tau = 21$; (b) $\alpha = 0.01, \beta = 0.001, \tau = 41$; (c) $\alpha = 0.1, \beta = 0.001, \tau = 41$; (d) $\alpha = 0.01, \beta = 0.01, \tau = 41$.

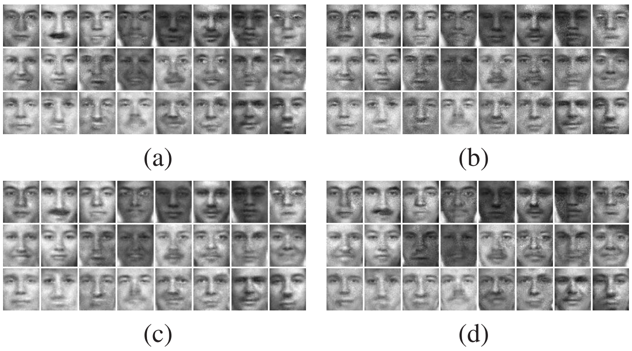


Figure 10. Some reconstructed images of FERET database using components of Fig. 9, with corresponding original images shown in Fig. 6.

we can see that τ affects the component size. That is, when τ is larger, the learned components are tend to be bigger in size. Second, Compare Fig. 9(c) ($\alpha = 0.1$) with Fig. 9(b) ($\alpha = 0.01$), it can be seen that α influences the smoothness of the learned components. The learned components are tend to be more smoothed when α is larger. Finally, compared to Fig. 9(b) ($\beta = 0.001$), Fig. 9(d) ($\beta = 0.01$) shows that β has an impact on the locality of the learned components, since larger β makes the learned components more localized. Besides, Fig. 10 demonstrates that the reconstruction is comparatively better with smaller values of α and β.

5. Summary

We have presented a GRF modeling based NMF algorithm for learning object parts. The formulation is based on an energy function with the prior knowledge of smoothness and locality. Using this technique of directly modeling on the structure of components, the GRF-NMF algorithm is able to learn sparse, smooth, and localized object parts. We have demonstrated that the GRF-NMF algorithm is easy to implement and fast to compute, because it is based on matrix-wise update and local filtering. We have also written a convergence proof for the derived algorithm. Experimental results on both synthesized and real image databases have shown that the proposed GRF-NMF algorithm significantly outperforms other NMF related algorithms in sparsity, smoothness, and locality of the learned components. Future works would be to apply the derived GRF-NMF algorithm for more complicated computer vision problems.

Acknowledgements.

This work was supported by the following funding resources: National Hi-Tech (863) Program Projects #2008AA01Z124, and AuthenMetric R&D Funds.

References

[1] D. Donoho and V. Stodden. "When does non-negative matrix factorization give a correct decomposition into parts". In *Proceedings of Neural Information Processing Systems*. MIT Press, 2003.

[2] P. O. Hoyer. "Non-negative sparse coding". In *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, pages 557–565, 2002.

[3] P. O. Hoyer and P. Dayan. "Non-negative matrix factorization with sparseness constraints". *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[4] D. D. Lee and H. S. Seung. "Learning the parts of objects by non-negative matrix factorization". *Nature*, 401:788–791, 1999.

[5] D. D. Lee and H. S. Seung. "Algorithms for non-negative matrix factorization". In *Proceedings of Neural Information Processing Systems*, volume 13, pages 556–562, 2001.

[6] S. Z. Li, X. W. Hou, and H. J. Zhang. "Learning spatially localized, parts-based representation". In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Hawaii, December 11-13 2001.

[7] W. Liu, N. Zheng, and X. Lu. "Non-negative matrix factorization for visual coding". In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 2003.

[8] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. "The FERET evaluation methodology for face-recognition algorithms". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[9] M. N. Schmidt and H. Laurberg. "Nonnegative matrix factorization with gaussian process priors". *Intell. Neuroscience*, 8(1):1–10, 2008.

[10] W. Zheng, S. Z. Li, J. Lai, and S. Liao. "On constrained sparse matrix factorization". In *Proceedings of IEEE International Conference on Computer Vision*, Oct. 2007.